

Здесь будет титульник, листай ниже

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ.....	5
1.1 Описание входных данных.....	6
1.2 Описание выходных данных.....	6
2 МЕТОД РЕШЕНИЯ.....	8
3 ОПИСАНИЕ АЛГОРИТМОВ.....	9
3.1 Алгоритм функции main.....	9
3.2 Алгоритм функции func.....	9
3.3 Алгоритм конструктора класса MyClass.....	10
3.4 Алгоритм конструктора класса MyClass.....	10
3.5 Алгоритм конструктора класса MyClass.....	11
3.6 Алгоритм деструктора класса MyClass.....	11
3.7 Алгоритм метода input класса MyClass.....	11
3.8 Алгоритм метода sum класса MyClass.....	12
3.9 Алгоритм метода method1 класса MyClass.....	12
3.10 Алгоритм метода method2 класса MyClass.....	13
4 БЛОК-СХЕМЫ АЛГОРИТМОВ.....	14
5 КОД ПРОГРАММЫ.....	21
5.1 Файл main.cpp.....	21
5.2 Файл MyClass.cpp.....	21
5.3 Файл MyClass.h.....	23
6 ТЕСТИРОВАНИЕ.....	24
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	25

1 ПОСТАНОВКА ЗАДАЧИ

Дан объект следующей конструкции:

В закрытом доступе имеется массив целого типа. Количество элементов массива четное и больше двух. Объект имеет функциональность:

- конструктор по умолчанию, в начале работы выдает сообщение;
- параметризованный конструктор, передается целочисленный параметр.

Параметр должен иметь значение больше 2 и быть четным. По значению параметра определяется размерность целочисленного массива из закрытой области. В начале работы выдает сообщение;

- метод деструктор, который выдает сообщение что он отработал;
- метод ввода данных для созданного массива;

- метод 1, который суммирует значения очередной пары элементов и сумму присваивает первому элементу пары. Далее суммирует элементы полученного массива и возвращает это значение. Например, пусть массив состоит из элементов {1,2,3,4}. В результате суммирования пар получим массив {3,2,7,4};

- метод 2, который умножает значения очередной пары элементов и результат присваивает первому элементу пары. Далее суммирует элементы полученного массива и возвращает это значение. Например, пусть массив состоит из элементов {1,2,3,4}. В результате умножения пар получим массив {2,2,12,4};

- метод который, суммирует значения элементов массива и возвращает это значение.

Разработать функцию, которая в качестве параметра получает объект по значению. Функция вызывается метод 2, далее выводит сумму элементов массива с новой строки.

В основной функции реализовать алгоритм:

1. Ввод размерности массива.

2. Вывод значения размерности массива.
3. Если размерность массива некорректная, вывод сообщения и завершить работу алгоритма.
4. Вывод значения размерности массива.
5. Создание объекта.
6. Вызов метода для ввода значений элементов массива.
7. Вызов функции передача в качестве аргумента объекта.
8. Вызов метода 1 от имени объекта.
9. Вывод суммы элементов массива объекта с новой строки.

Разработать конструктор копии объекта, для корректного выполнения вычислений. В начале работы конструктор копии выдает сообщение с новой строки.

1.1 Описание входных данных

Первая строка:

«Целое число»

Вторая строка:

«Целое число» «Целое число»

Пример.

8

1 2 3 4 5 6 7 8

1.2 Описание выходных данных

Если введенная размерность массива допустима, то в первой строке выводится это значение:

«Целое число»

Если введенная размерность массива не больше двух или нечетная, то в первой строке выводится некорректное значение и вопросительный знак:

«Целое число»?

Конструктор по умолчанию в начале работы с новой строки выдает сообщение:

Default constructor

Параметризованный конструктор в начале работы с новой строки выдает сообщение:

Constructor set

Конструктор по умолчанию в начале работы с новой строки выдает сообщение:

Copy constructor

Деструктор в начале работы с новой строки выдает сообщение:

Destructor

Пример вывода.

8

Constructor set

Copy constructor

120

Destructor

56

Destructor

2 МЕТОД РЕШЕНИЯ

Для решения задачи используется:

Объекты стандартных потоков ввода/вывода cin, cout;

Условный оператор;

Оператор цикла со счетчиком;

Функция:

Func;

Класс MyClass:

Деструктор:

~MyClass;

Конструкторы:

MyClass;

MyClass;

MyClass;

Методы:

input;

sum;

method1;

method2.

3 ОПИСАНИЕ АЛГОРИТМОВ

Согласно этапам разработки, после определения необходимого инструментария в разделе «Метод», составляются подробные описания алгоритмов для методов классов и функций.

3.1 Алгоритм функции main

Функционал: отсутствует.

Параметры: отсутствуют.

Возвращаемое значение: целое - признак успеха.

Алгоритм функции представлен в таблице 1.

Таблица 1 – Алгоритм функции main

№	Предикат	Действия	№ перехода
1		Создание целочисленной переменной size	2
2		Ввод size	3
3	size<=2 или size нечетное	вывод сообщения об ошибке	∅
			4
4		вывод size	5
5		создание obj класса MyClass	6
6		Вызов input	7
7		Вызов func	8
8		Вызов method2	9
9		Вызов sum	∅

3.2 Алгоритм функции func

Функционал: копирование.

Параметры: MyClass obj.

Возвращаемое значение: отсутствует.

Алгоритм функции представлен в таблице 2.

Таблица 2 – Алгоритм функции func

№	Предикат	Действия	№ перехода
1		Вызов method1	2
2		Вызов sum	∅

3.3 Алгоритм конструктора класса MyClass

Функционал: создание объекта.

Параметры: отсутствует.

Алгоритм конструктора представлен в таблице 3.

Таблица 3 – Алгоритм конструктора класса MyClass

№	Предикат	Действия	№ перехода
1		Вывод Default constructor	∅

3.4 Алгоритм конструктора класса MyClass

Функционал: создает объект.

Параметры: size.

Алгоритм конструктора представлен в таблице 4.

Таблица 4 – Алгоритм конструктора класса MyClass

№	Предикат	Действия	№ перехода
1		Вывод Constructor set	2
2		Создание целочисленного массива размера size	∅

3.5 Алгоритм конструктора класса MyClass

Функционал: создает объект.

Параметры: MyClass& obj.

Алгоритм конструктора представлен в таблице 5.

Таблица 5 – Алгоритм конструктора класса MyClass

№	Предикат	Действия	№ перехода
1		Вывод Copy constructor	2
2		копирование значений переменных obj	∅

3.6 Алгоритм деструктора класса MyClass

Функционал: удаляет объект.

Параметры: отсутствует.

Алгоритм деструктора представлен в таблице 6.

Таблица 6 – Алгоритм деструктора класса MyClass

№	Предикат	Действия	№ перехода
1		Вывод Destructor	∅

3.7 Алгоритм метода input класса MyClass

Функционал: Ввод чисел в массив.

Параметры: отсутствует.

Возвращаемое значение: отсутствует.

Алгоритм метода представлен в таблице 7.

Таблица 7 – Алгоритм метода *input* класса *MyClass*

№	Предикат	Действия	№ перехода
1		создание целочисленной переменной <i>n</i>	2
2	<i>i</i> < <i>size</i>		3
			∅
3		Ввод <i>n</i>	4
4		<i>arr</i> [<i>i</i>]:= <i>n</i>	5
5		<i>i</i> ++	2

3.8 Алгоритм метода *sum* класса *MyClass*

Функционал: суммирование элементов массива.

Параметры: отсутствует.

Возвращаемое значение: отсутствует.

Алгоритм метода представлен в таблице 8.

Таблица 8 – Алгоритм метода *sum* класса *MyClass*

№	Предикат	Действия	№ перехода
1		создание целочисленной переменной <i>s</i>	2
2	<i>i</i> < <i>size</i>		3
			5
3		<i>s</i> := <i>arr</i> [<i>i</i>]+ <i>s</i>	4
4		<i>i</i> ++	2
5		Вывод <i>s</i>	∅

3.9 Алгоритм метода *method1* класса *MyClass*

Функционал: Умножает нечетный элемент массива на соседним четный.

Параметры: отсутствует.

Возвращаемое значение: отсутствует.

Алгоритм метода представлен в таблице 9.

Таблица 9 – Алгоритм метода *method1* класса *MyClass*

№	Предикат	Действия	№ перехода
1	i < size		2
			∅
2		arr[i]:=arr[i]+arr[i+1]	3
3		i++	1

3.10 Алгоритм метода *method2* класса *MyClass*

Функционал: Складывает нечетный элемент массива с соседним четным.

Параметры: отсутствует.

Возвращаемое значение: отсутствует.

Алгоритм метода представлен в таблице 10.

Таблица 10 – Алгоритм метода *method2* класса *MyClass*

№	Предикат	Действия	№ перехода
1	i < size		2
			∅
2		arr[i]:=arr[i]+arr[i+1]	3
3		i++	1

4 БЛОК-СХЕМЫ АЛГОРИТМОВ

Представим описание алгоритмов в графическом виде на рисунках 1-7.

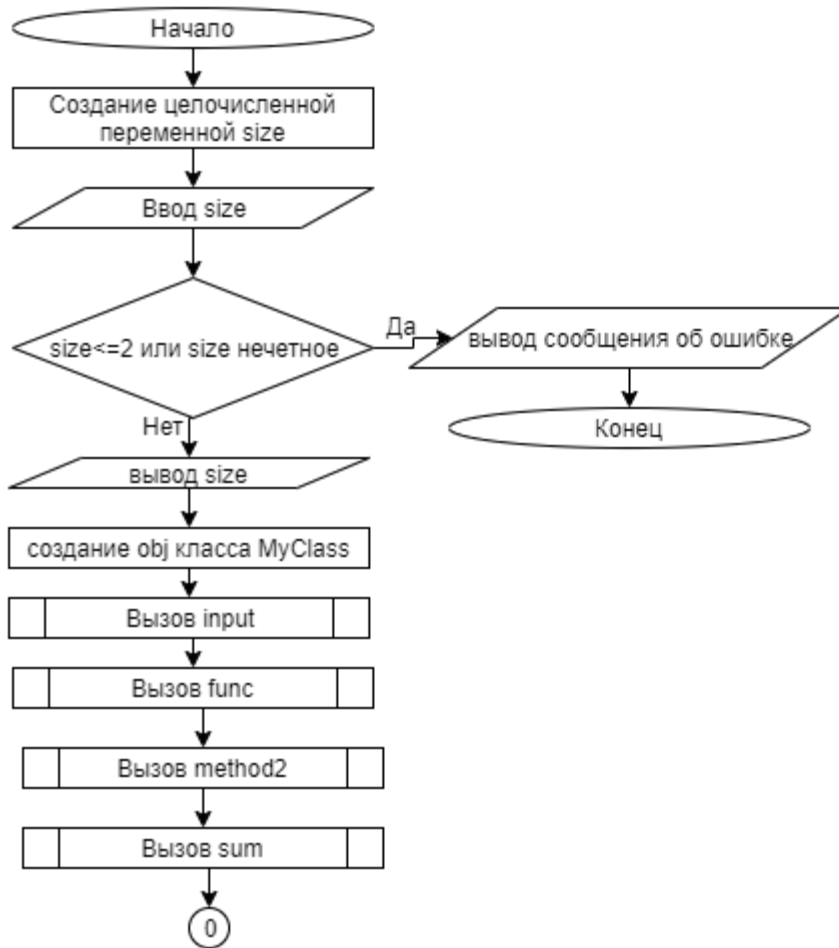


Рисунок 1 – Блок-схема алгоритма

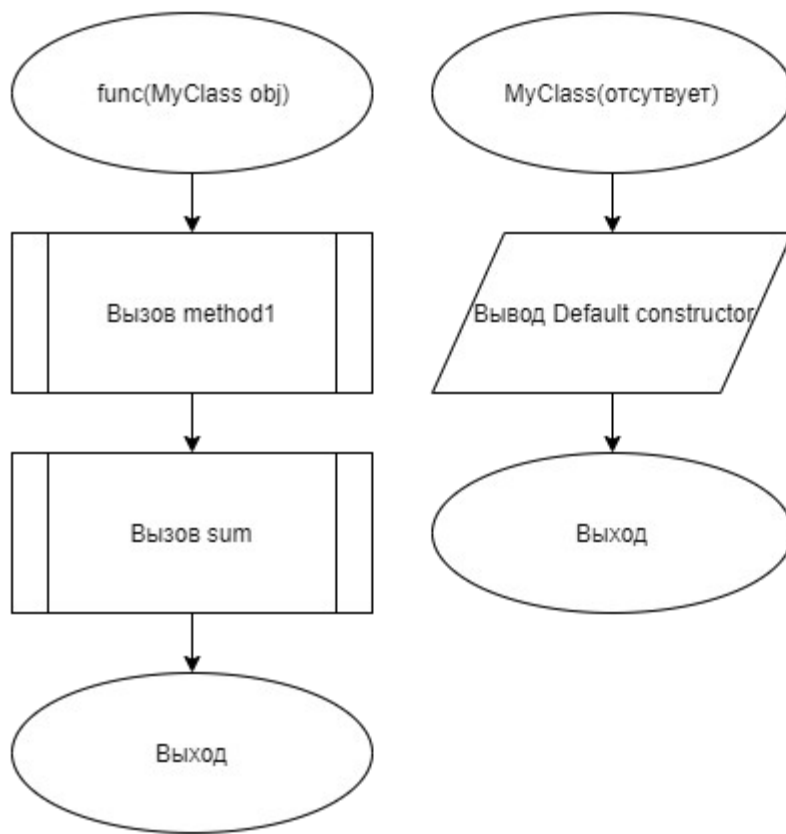


Рисунок 2 – Блок-схема алгоритма

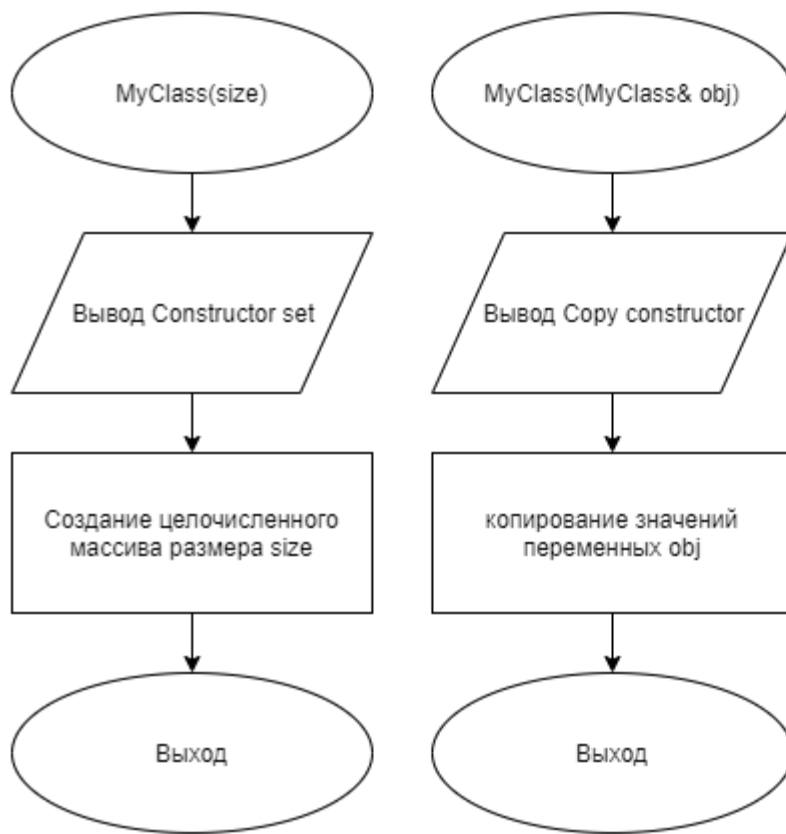


Рисунок 3 – Блок-схема алгоритма

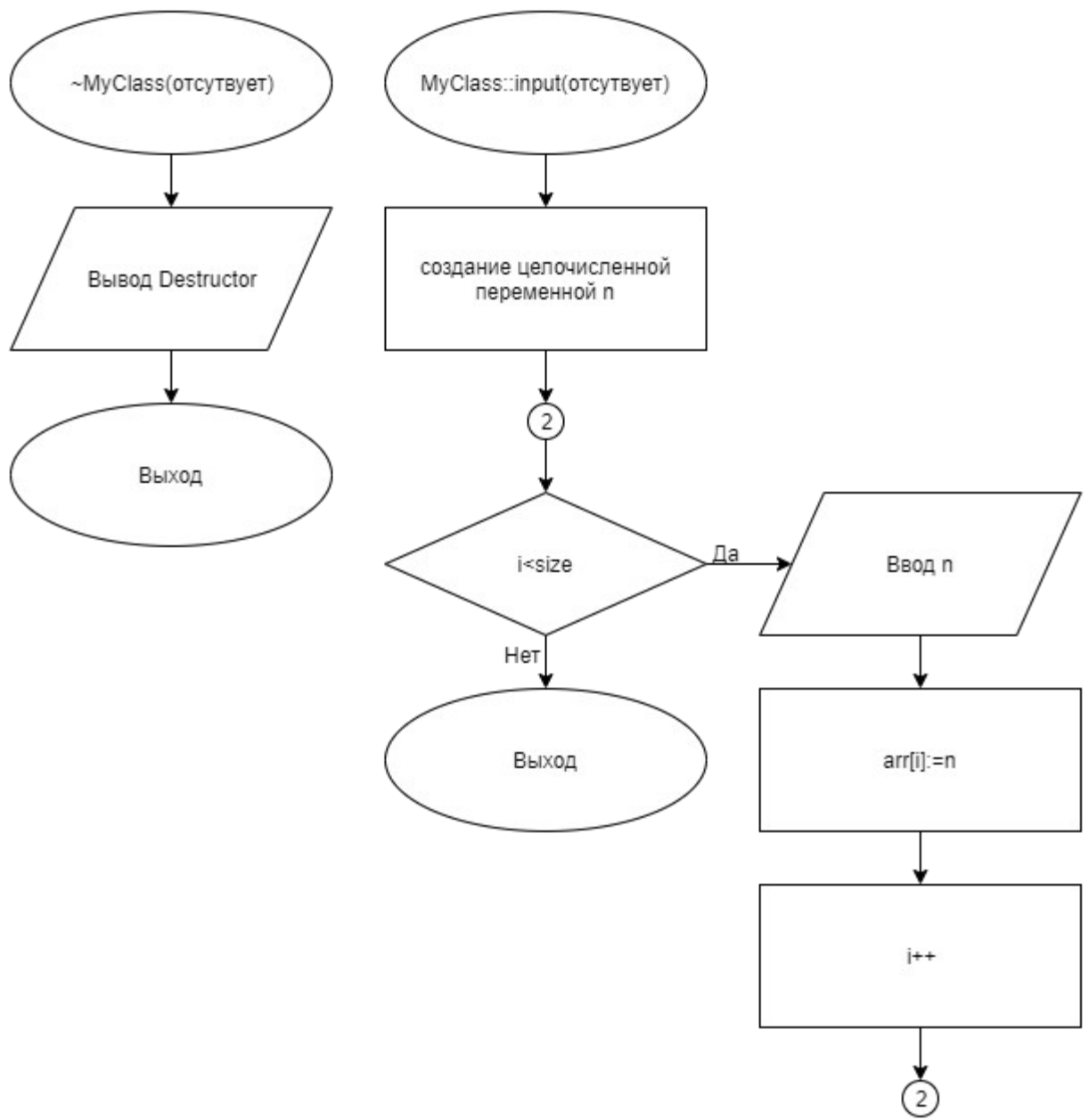


Рисунок 4 – Блок-схема алгоритма

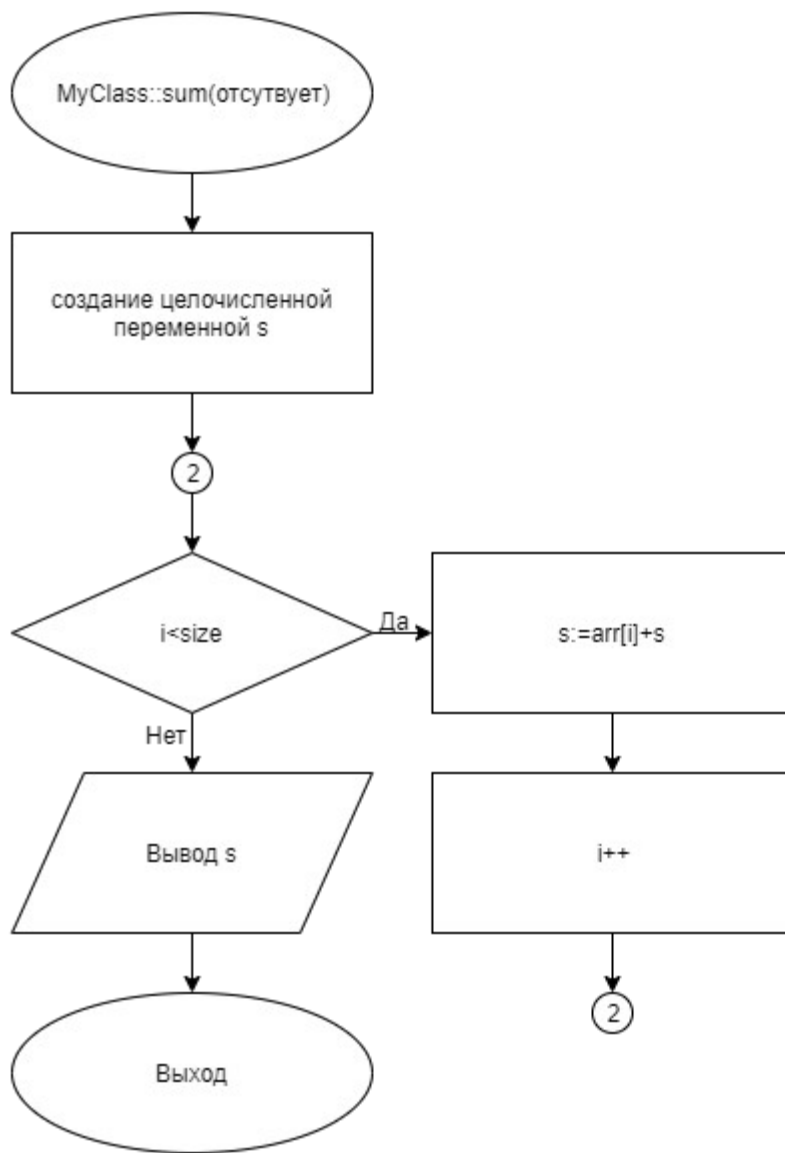


Рисунок 5 – Блок-схема алгоритма

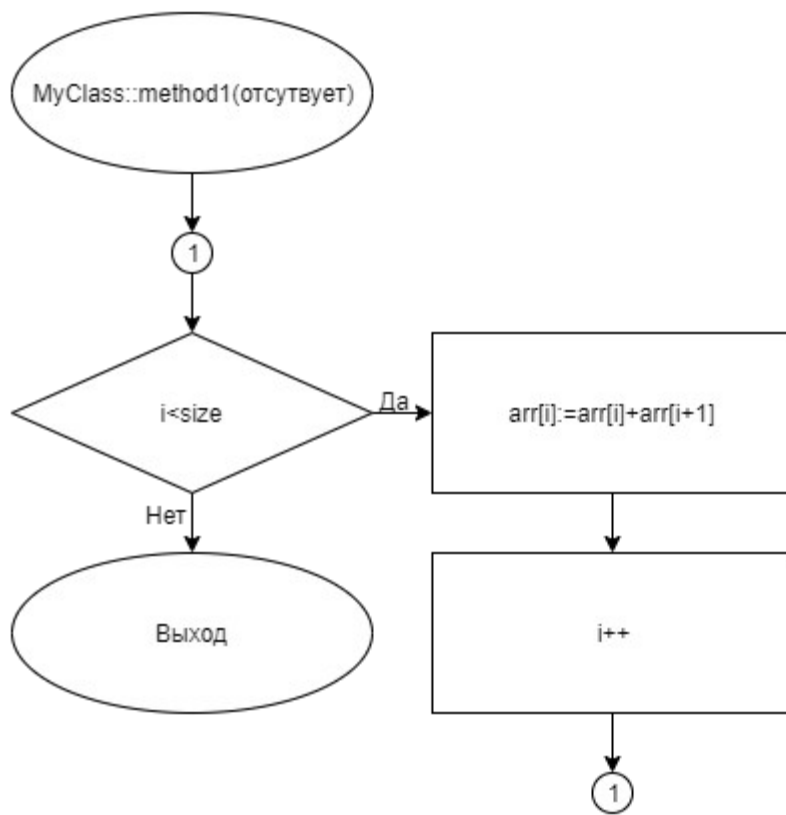


Рисунок 6 – Блок-схема алгоритма

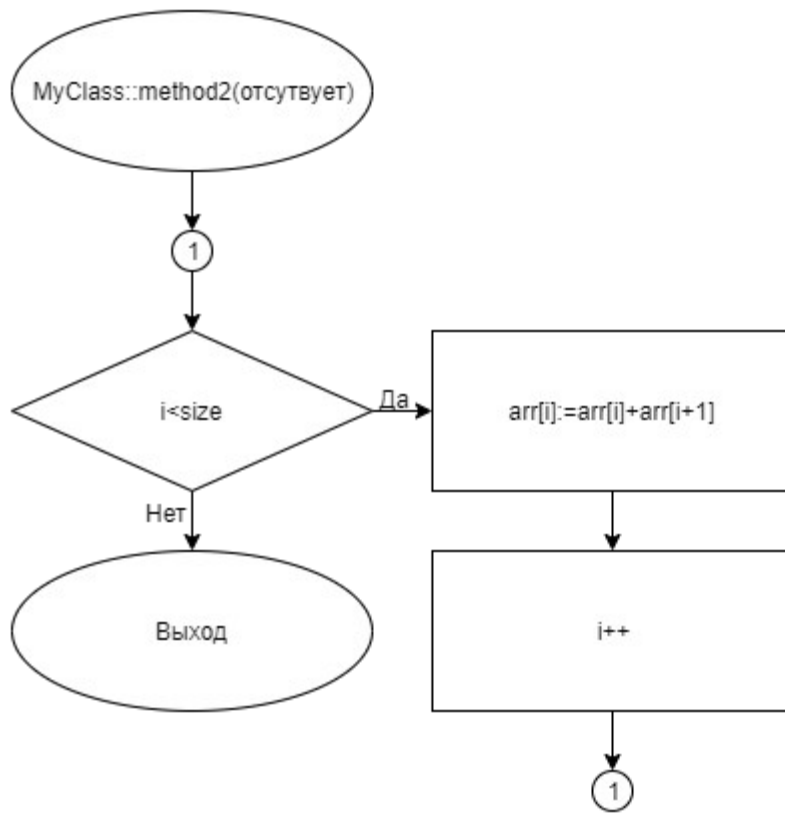


Рисунок 7 – Блок-схема алгоритма

5 КОД ПРОГРАММЫ

Программная реализация алгоритмов для решения задачи представлена ниже.

5.1 Файл main.cpp

Листинг 1 – main.cpp

```
#include "MyClass.h"
#include<iostream>
using namespace std;

void func(MyClass obj)
{
    obj.method1();
    obj.sum();
}

int main()
{
    int size;
    cin>>size;
    if ((size<=2) ||(size%2!=0))
    {
        cout<<size<<"?";
        return 0;
    }
    cout<<size<<endl;
    MyClass obj(size);
    obj.input();
    func(obj);
    cout<<endl;
    obj.method2();
    obj.sum();
    return(0);
}
```

5.2 Файл MyClass.cpp

Листинг 2 – MyClass.cpp

```
#include "MyClass.h"
#include<iostream>
#include<string>
using namespace std;
```

```

MyClass::MyClass()
{
    cout<<"Default constructor"<<endl;
}
MyClass::MyClass(int size)
{
    arr=new int [size];
    cout<<"Constructor set"<<endl;
    this->size=size;
}
MyClass::MyClass(const MyClass & obj)
{
    cout<<"Copy constructor"<<endl;
    size=obj.size;
    arr=new int[size];
    for(int i=0;i<size;i++)
        arr[i]=obj.arr[i];
}
void MyClass::input()
{
    int n;
    for(int i=0;i<size;i++)
    {
        cin>>n;
        arr[i]=n;
    }
}
/*void MyClass::viod()
{
    for (int i=0;i<size;i++)
    {
        cout<<arr[i]<<" ";
    }
    cout<<endl;
}*/
MyClass::~MyClass()
{
    cout<<"Destructor";
    if (arr!=nullptr)
        delete[]arr;
}
void MyClass::method1()
{
    for (int i=0;i<size;i=i+2)
    {
        arr[i]=arr[i]*arr[i+1];
    }
}
void MyClass::method2()
{
    for (int i=0;i<size;i=i+2)
    {
        arr[i]=arr[i]+arr[i+1];
    }
}
void MyClass::sum()

```

```
{
    int s=0;
    for (int i=0;i<size;i++)
    {
        s=s+arr[i];
    }
    cout<<s<<endl;
}
```

5.3 Файл MyClass.h

Листинг 3 – MyClass.h

```
#ifndef __MYCLASS_H__
#define __MYCLASS_H__
#include<iostream>
using namespace std;

class MyClass
{
private:
    int* arr;
    int size;
public:
    ~MyClass();
    MyClass();
    MyClass(int size);
    MyClass(const MyClass& obj);
    void input();
    void vivod();
    void sum();
    void method1();
    void method2();
};
#endif
```

6 ТЕСТИРОВАНИЕ

Результат тестирования программы представлен в таблице 11.

Таблица 11 – Результат тестирования программы

Входные данные	Ожидаемые выходные данные	Фактические выходные данные
8 1 2 3 4 5 6 7 8	8 Constructor set Copy constructor 120 Destructor 56 Destructor	8 Constructor set Copy constructor 120 Destructor 56 Destructor
1 1	1?	1?

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Васильев А.Н. Объектно-ориентированное программирование на C++. Издательство: Наука и Техника. Санкт-Петербург, 2016г. 543 стр.
2. Шилдт Г. C++: базовый курс. 3-е изд. Пер. с англ.. — М.: Вильямс, 2017. — 624 с.
3. Методическое пособие для проведения практических заданий, контрольных и курсовых работ по дисциплине «Объектно-ориентированное программирование» [Электронный ресурс] – URL: https://mirea.aco-avrora.ru/student/files/methodicheskoe_posobie_dlya_laboratornyh_rabot_3.pdf (дата обращения 05.05.2021).
4. Приложение к методическому пособию студента по выполнению заданий в рамках курса «Объектно-ориентированное программирование» [Электронный ресурс]. URL: https://mirea.aco-avrora.ru/student/files/Prilozheniye_k_methodichke.pdf (дата обращения 05.05.2021).
5. Видео лекции по курсу «Объектно-ориентированное программирование» [Электронный ресурс]. АСО «Аврора».
6. Антик М.И. Дискретная математика [Электронный ресурс]: Учебное пособие /Антик М.И., Казанцева Л.В. — М.: МИРЭА — Российский технологический университет, 2018 — 1 электрон. опт. диск (CD-ROM).